

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Methods and Systems for Processing Media Content

Inventor(s):
Jason McCartney
Keith Toussaint

ATTORNEY'S DOCKET NO. msl-784us

1 **TECHNICAL FIELD**

2 This invention relates to methods and systems for processing media
3 content.

4

5 **BACKGROUND**

6 With the technological advance of computers and the software that runs on
7 computers, users are now able to enjoy many features which, just a few years ago,
8 did not exist. For example, users can now play various media and multimedia
9 content on their personal or laptop computers, thus providing an improved user
10 experience. For example, most computers today are able to play compact discs
11 (CDs) so that a user can listen to a favorite artist or artists while working on their
12 computer. Additionally, many computers are equipped with a digital versatile disc
13 (DVD) drive so that the user can watch movies on their personal computer.

14 As users become more used to advanced features on their computers, such
15 as those mentioned above, their expectations of the various additional innovative
16 features will undoubtedly continue to grow. For example, consider a media player
17 software application that enables a user to play a CD on their computer. Typical
18 applications allow a user to display, via the use of a mouse, the track information
19 that is associated with the CD by clicking on an appropriate user interface (UI).
20 Such track information typically includes track numbers, song titles, and playing
21 times. As consumers become more sophisticated, however, this paradigm is going
22 to be perceived as inadequate.

23 Accordingly, this invention arose out of concerns associated with providing
24 improved systems and methods for processing media content that provide an
25 improved, rich and robust user experience.

1

2 **SUMMARY**

3 Methods and systems are described that greatly enhance the user experience
4 when playing various media such as CDs and DVDs. One or more databases,
5 managed by a server, maintain metadata associated with various media. The
6 metadata can include any type of additional information that can be of interest to a
7 user or consumer of the media. Through inventive mapping techniques, physical
8 IDs that are associated with a user's specific media are mapped to logical IDs.
9 The logical IDs then serve as the basis for database queries that retrieve the
10 metadata. The retrieved metadata can then be sent to a client computer and
11 displayed for or otherwise consumed by the user. In one embodiment, a set of
12 extensible markup language (XML) schema are provided for data exchange
13 between the client and the server.

14 Various embodiments also provide an opportunity for user feedback to be
15 immediately incorporated into a server so that it can be used to provide an
16 enriched user experience for others. Specifically, users can provide their own
17 physical ID to logical ID mappings which then become available for use in
18 retrieving the metadata. Security measures can also provide an added degree of
19 security to ensure that the user-provided data can be trusted.

20 Other embodiments provide a Wizard user interface (UI) that can be used to
21 assist in mapping physical IDs to logical IDs. Specifically, in the event that a
22 logical ID is not initially located for a corresponding physical ID associated with
23 the user's media, the Wizard can guide the user through an information collection
24 process directed to identifying the user's specific media. This is done by
25 collecting information from the user pertaining to their specific media, and

1 searching for matches for the media. This discovery process can be iterative in
2 nature and can result in multiple exchanges between the user and the Wizard. If
3 the Wizard is successful in identifying a user's specific media, then a physical ID
4 to logical ID mapping is created for that media. This mapping can then be used as
5 the basis for metadata database queries. The Wizard can also enable a user to
6 create a physical ID to logical ID mapping for their media in the event that it is
7 unable to identify the user's media.

8

9 **BRIEF DESCRIPTION OF THE DRAWINGS**

10 Fig. 1 is a diagram that illustrates an exemplary system in which various
11 embodiments can be implemented.

12 Fig. 2 is a block diagram that illustrates a computing environment in which
13 various embodiments can be implemented.

14 Fig. 3 is a diagram that illustrates aspects of physical IDs and logical IDs in
15 accordance with one embodiment.

16 Fig. 4 is a flow diagram that illustrates some general inventive concepts.

17 Fig. 5 is a block diagram that illustrates a canonical table and a user-
18 provided mappings table in accordance with one embodiment.

19 Fig. 6 is a flow diagram that illustrates steps in a method in accordance
20 with one embodiment.

21 Fig. 7 is a flow diagram that illustrates steps in a method in accordance
22 with one embodiment.

23 Figs. 8-12 illustrate an exemplary user interface (UI) that is provided by a
24 Wizard in accordance with one embodiment.

1 Fig. 13 is a flow diagram that illustrates steps in a method in accordance
2 with one embodiment.

3 Fig. 14 is a flow diagram that illustrates steps in a method in accordance
4 with one embodiment.

5 Fig. 15 is a block diagram that illustrates a user feedback table in
6 accordance with one embodiment.

7 Fig. 16 is a flow diagram that illustrates steps in a method in accordance
8 with one embodiment.

9 Fig. 17 is a block diagram that illustrates an exemplary process for ensuring
10 the integrity of user-provided physical ID to logical ID mappings.

11

12 **DETAILED DESCRIPTION**

13 **Overview**

14 The embodiments described below provide methods and systems that
15 enable a user or, more accurately, an enabled media player that is executing on a
16 computing device or client, to access, retrieve, and display for a user, so-called
17 metadata that is associated with specific media content that is being played on the
18 media player. This is done by matching the metadata to the specific media content
19 that is being experienced by a user, and returning the metadata to the user's
20 computing device. In the examples that are given below, the media content is
21 described in the context of content that is embodied on a CD or a DVD. It is to be
22 appreciated and understood that the media content can be embodied on any
23 suitable media, and that the specific examples described herein are given for
24 purposes that include assisting the reader in understanding the inventive
25 principles. For example, the media content can include specially encoded media

1 content in the form of, for example, an encoded media file. An example of such
2 specially encoded media content can include, without limitation, media content
3 encoded in Windows Media format using the Windows Media Player.

4 Various features of the described systems and methods include a set of
5 databases, client side executable code, and a series of server side processes that
6 provide for querying and maintaining the databases. One logical organization of
7 an exemplary system includes the following: (1) a process to map a piece of
8 physical media to a unique database key or, as referred to herein, a “logical ID”,
9 (2) a query process to retrieve information from a database based on the unique
10 database key or logical ID, (3) a data return mechanism and schema set that is
11 used to return data, (4) a user feedback system that allows users to contribute to
12 the set of understood keys or logical IDs, and, (5) a set of management processes
13 that process user contributions, ensure the integrity of the contributions and make
14 the contributions available to the rest of the process.

15 The resultant system provides a user with the ability to place a CD or DVD
16 into an enabled media playing device (e.g. a computer running Microsoft
17 Windows and Windows Media Player) and expect not only to experience the
18 media contained on the disc, but also have access to all manner of related metadata
19 (e.g. cover art, performer biographies, reviews, related performers, where to buy
20 similar items, upcoming concerts, ticket sales, URLs to other related experiences
21 including buying experiences, and the like). In addition, the user community has
22 the ability to contribute key information to the process to improve the experience
23 for other users.

24 Fig. 1 shows an exemplary system in which the embodiments described
25 below can be implemented. The system includes one or more client computers—

1 here 100, 102, a network 104, one or more server computers 106, and one or more
2 databases 108. An exemplary network comprises the Internet, although any
3 suitable network can be used.

4 In this system, a user on the client side inserts their media into their
5 computer, or otherwise causes the content on the media to be experienced. The
6 media is identified by a physical ID that is used to access a logical ID that
7 uniquely identifies the media. The logical ID is then used as the basis for
8 metadata queries of database 108. These queries are designed to retrieve a rich set
9 of related metadata for the user. The metadata is then returned to the client via
10 network 104 and displayed for the user.

11 The description below will provide detailed aspects of the above systems
12 and various methods that all contribute to a much richer user experience.

13

14 **Exemplary Computer System**

15 Fig. 2 illustrates an example of a suitable computing environment 200 on
16 which the inventive systems and methods described below can be implemented.

17 It is to be appreciated that computing environment 200 is only one example
18 of a suitable computing environment and is not intended to suggest any limitation
19 as to the scope of use or functionality of the inventive embodiments described
20 below. Neither should the computing environment 200 be interpreted as having
21 any dependency or requirement relating to any one or combination of components
22 illustrated in the exemplary computing environment 200.

23 The inventive techniques can be operational with numerous other general
24 purpose or special purpose computing system environments or configurations.
25 Examples of well known computing systems, environments, and/or configurations

1 that may be suitable for use with the inventive techniques include, but are not
2 limited to, personal computers, server computers, thin clients, thick clients, hand-
3 held or laptop devices, multiprocessor systems, microprocessor-based systems, set
4 top boxes, programmable consumer electronics, network PCs, minicomputers,
5 mainframe computers, distributed computing environments that include any of the
6 above systems or devices, and the like.

7 In certain implementations, the inventive techniques can be described in the
8 general context of computer-executable instructions, such as program modules,
9 being executed by a computer. Generally, program modules include routines,
10 programs, objects, components, data structures, etc. that perform particular tasks
11 or implement particular abstract data types. The inventive techniques may also be
12 practiced in distributed computing environments where tasks are performed by
13 remote processing devices that are linked through a communications network. In
14 a distributed computing environment, program modules may be located in both
15 local and remote computer storage media including memory storage devices.

16 In the illustrated example, computing system 200 comprises one or more
17 processors or processing units 202, a system memory 204, and a bus 206 that
18 couples various system components including the system memory 204 to the
19 processor 202.

20 Bus 206 is intended to represent one or more of any of several types of bus
21 structures, including a memory bus or memory controller, a peripheral bus, an
22 accelerated graphics port, and a processor or local bus using any of a variety of
23 bus architectures. By way of example, and not limitation, such architectures
24 include Industry Standard Architecture (ISA) bus, Micro Channel Architecture
25 (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association

1 (VESA) local bus, and Peripheral Component Interconnects (PCI) buss also
2 known as Mezzanine bus.

3 Computer 200 typically includes a variety of computer readable media.
4 Such media may be any available media that is locally and/or remotely accessible
5 by computer 200, and it includes both volatile and non-volatile media, removable
6 and non-removable media.

7 In Fig. 2, the system memory 204 includes computer readable media in the
8 form of volatile, such as random access memory (RAM) 210, and/or non-volatile
9 memory, such as read only memory (ROM) 208. A basic input/output system
10 (BIOS) 212, containing the basic routines that help to transfer information
11 between elements within computer 200, such as during start-up, is stored in ROM
12 208. RAM 210 typically contains data and/or program modules that are
13 immediately accessible to and/or presently be operated on by processing unit(s)
14 202.

15 Computer 200 may further include other removable/non-removable,
16 volatile/non-volatile computer storage media. By way of example only, Fig. 2
17 illustrates a hard disk drive 228 for reading from and writing to a non-removable,
18 non-volatile magnetic media (not shown and typically called a “hard drive”), a
19 magnetic disk drive 230 for reading from and writing to a removable, non-volatile
20 magnetic disk 232 (e.g., a “floppy disk”), and an optical disk drive 234 for reading
21 from or writing to a removable, non-volatile optical disk 236 such as a CD-ROM,
22 DVD-ROM or other optical media. The hard disk drive 228, magnetic disk drive
230, and optical disk drive 234 are each connected to bus 206 by one or more
24 interfaces 226.

1 The drives and their associated computer-readable media provide
2 nonvolatile storage of computer readable instructions, data structures, program
3 modules, and other data for computer 200. Although the exemplary environment
4 described herein employs a hard disk 228, a removable magnetic disk 232 and a
5 removable optical disk 236, it should be appreciated by those skilled in the art that
6 other types of computer readable media which can store data that is accessible by a
7 computer, such as magnetic cassettes, flash memory cards, digital video disks,
8 random access memories (RAMs), read only memories (ROM), and the like, may
9 also be used in the exemplary operating environment.

10 A number of program modules may be stored on the hard disk 228,
11 magnetic disk 232, optical disk 236, ROM 208, or RAM 210, including, by way of
12 example, and not limitation, an operating system 214, one or more application
13 programs 216 (e.g., media player 224), other program modules 218, and program
14 data 220. Some of the application programs can be configured to present a user
15 interface (UI) that is configured to allow a user to interact with the application
16 program in some manner using some type of input device. This UI is typically a
17 visual display that is capable of receiving user input and processing that user input
18 in some way. Such a UI may, for example, comprises one or more buttons or
19 controls that can be clicked on by a user. Media player application 224 can be any
20 suitable media player application that is configured to play any suitable media so
21 that a user can experience the content that is embodied on the media. Two
22 exemplary media player applications can include a CD media player application
23 and a DVD media player application.

24 Continuing with Fig. 2, a user may enter commands and information into
25 computer 200 through input devices such as keyboard 238 and pointing device 240

1 (such as a “mouse”). Other input devices may include a audio/video input
2 device(s) 253, a microphone, joystick, game pad, satellite dish, serial port, scanner,
3 or the like (not shown). These and other input devices are connected to the
4 processing unit(s) 202 through input interface(s) 242 that is coupled to bus 206,
5 but may be connected by other interface and bus structures, such as a parallel port,
6 game port, or a universal serial bus (USB).

7 A monitor 256 or other type of display device is also connected to bus 206
8 via an interface, such as a video adapter 244. In addition to the monitor, personal
9 computers typically include other peripheral output devices (not shown), such as
10 speakers and printers, which may be connected through output peripheral interface
11 246.

12 Computer 200 may operate in a networked environment using logical
13 connections to one or more remote computers, such as a remote computer 250.
14 Remote computer 250 may include many or all of the elements and features
15 described herein relative to computer 200.

16 As shown in Fig. 2, computing system 200 can be communicatively
17 coupled to remote devices (e.g., remote computer 250) through a local area
18 network (LAN) 251 and a general wide area network (WAN) 252. Such
19 networking environments are commonplace in offices, enterprise-wide computer
20 networks, intranets, and the Internet.

21 When used in a LAN networking environment, the computer 200 is
22 connected to LAN 251 through a suitable network interface or adapter 248. When
23 used in a WAN networking environment, the computer 200 typically includes a
24 modem 254 or other means for establishing communications over the WAN 252.

1 The modem 254, which may be internal or external, may be connected to the
2 system bus 206 via the user input interface 242, or other appropriate mechanism.

3 In a networked environment, program modules depicted relative to the
4 personal computer 200, or portions thereof, may be stored in a remote memory
5 storage device. By way of example, and not limitation, Fig. 2 illustrates remote
6 application programs 216 as residing on a memory device of remote computer
7 250. It will be appreciated that the network connections shown and described are
8 exemplary and other means of establishing a communications link between the
9 computers may be used.

10

11 **Physical Media Identification and Unique Logical ID Mapping**

12 In one described embodiment, a physical ID or “PID” is associated with
13 each media upon which the content that is to be experienced by a user resides.
14 The PID is assigned or otherwise associated with a logical ID or “LID”, and the
15 LID is then used as the basis for any database queries.

16 Consider, for example, Fig. 3. There, six CDs are shown—two each of the
17 Backstreet Boys “Black and Blue” CD, Britney Spears “Stronger” CD and Weird
18 Al’s “Running with Sissors” CD. Each of these CDs belongs to a different person.
19 As shown, each CD has a physical ID associated with it. Each physical ID is
20 different. For example, there are two different physical IDs associated with the
21 Backstreet Boys CD (i.e. “12345” and “34567”). Yet, each of these different
22 physical IDs is mapped to the same logical ID (i.e. ABCDE). This logical ID is
23 then used by the system as the basis for any database queries for metadata
24 associated with the Backstreet Boys CD.

With respect to the physical IDs that are associated with the media, any suitable method or technique of generating a physical ID can be used. For example, when a user inserts a piece of media into a properly configured and enabled device, a piece of software code can execute and read data from the physical media. The software code can then compose a unique or nearly unique physical ID from that data.

In the case where the media comprises a CD, the software code can read the offsets (in frames, which have a resolution of 1/72nd of a second) of each track on the disc. A composite key or physical ID is then built and comprises a string of the hex values of these offsets, prefaced by a number of tracks on the disc and finished with a representation of the total length of the disc.

In the case where the media comprises a DVD, the software code can read the first 64 kilobytes of two files that are guaranteed to be on every DVD. These files are VIDEO_TS.IFO and VTS_01_0.IFO. The former contains main-menu information (VMGI), and the latter contains title set information (VTSI) for the first title on the DVD. After the appropriate data blocks are read, the code generates a 64-bit CRC (cyclic redundancy code) checksum of the data, resulting in an appropriately unique key or physical ID.

Of course, it is to be understood that the above two examples are simply two ways that a physical ID can be generated for two different types of media. Other methods of generating physical IDs, as well as other media types can be employed.

Calculation of the physical IDs takes place, in this example, on the client side by software code that executes on the client. Such code can comprise part of

1 a software-implemented media player that is configured to play the media of
2 interest.

3 Once the physical IDs are generated, the physical IDs are sent, via a
4 network such as the Internet and using a suitable protocol, for processing that is
5 described in detail below. To assist in understanding, at a more global level, the
6 processing that takes place, including generation of the physical IDs, the reader is
7 referred to Fig. 4. Fig. 4 is a work flow diagram that is useful in understanding the
8 processing that takes place on and between the client and the server in accordance
9 with one embodiment. More detailed discussion on certain aspects of the Fig. 4
10 diagram appears in other sections below.

11 At 400 a user inserts a particular media into an enabled media player and, at
12 402, a physical ID is generated for the particular piece of media. Examples of
13 how physicals IDs can be generated are given above. This physical ID is then
14 bundled up and sent to a server system for processing. This bundling can be done
15 in any suitable way using any suitable protocols. In one example, the physical ID
16 is passed, through an HTTP URL, to the server system. The server system can be
17 configured in any suitable way. In this example, the server can comprise active
18 server pages (ASP) code residing thereon running Microsoft's Internet Information
19 Server product. The code can also include a mechanism for converting the ASP
20 request into a Microsoft SQL Server XML query request, as will be understood by
21 those of skill in the art.

22 The physical ID is then used to query a lookup table 404 to determine
23 whether there is a proper logical ID associated with it. The logical ID represents
24 the piece of media in a metadata store or database 406. If there is a logical ID
25 associated with the physical ID, then that logical ID serves as a basis for a query

1 of database 406. This query then returns, to the user, metadata associated with the
2 user's media. This metadata can comprise a rich collection of data, with non-
3 limiting examples being given above.

4 If, on the other hand, no logical ID can be found for the user's physical ID,
5 then a Wizard 408 is presented to the user, on the client side, to attempt to find or
6 establish a logical ID for the user's piece of media or, more accurately, the
7 physical ID that is associated with the user's piece of media. For example, assume
8 that a user starts playing a CD that has a physical ID that has not yet been
9 processed by the system. When the server attempts to look up a logical ID
10 associated with the media's physical ID, no corresponding logical ID will be
11 found. Accordingly, the server then presents the Wizard to the user and attempts
12 to identify the user's media. The reason that the Wizard attempts to identify the
13 user's media is that there may already exist a logical ID that is associated with the
14 media. Recall that the same CD (e.g. U2's newest CD) may have several different
15 physical IDs associated with it, yet there will be only one logical ID to which all
16 of these physical IDs are mapped. If the system has not yet processed the user's
17 physical ID, it will seek to establish an association between that physical ID and
18 the logical ID that already exists in the system for that particular CD.

19 One example, of how this is done is given below in the section entitled
20 "Wizard". If the Wizard is successful in identifying the user's media (e.g. if the
21 user is successful in identifying their media using the Wizard), and there exists a
22 logical ID for the user's media, the server will establish, at 410, a physical ID to
23 logical ID mapping for the user's specific physical ID. The mapping will then
24 map the user's specific physical ID to the logical ID that is associated with the
25

1 user's media. This association is stored on a database 412 that contains physical
2 ID to logical ID mappings.

3 If the Wizard 408 is unsuccessful in identifying the user's particular media,
4 then the user can provide user-entered data at 414 that is then used to identify the
5 media for the system. The user can not only enter their own data for their media
6 (such as title, tracks and artist), but they can also establish a physical ID to logical
7 ID mapping for that media. This new logical ID will thus serve as the logical ID
8 for all subsequent physical IDs that are associated with that media. Consider, for
9 example, a situation in which a particular user is the first system user to play a
10 new CD. In this case, the system may not include a logical ID for the new
11 physical media. Accordingly, the first user then, through the Wizard, can be
12 prompted to enter any relevant information for the CD (i.e. title, artist, tracks,
13 track titles and the like), as well as a logical ID for the media so that an association
14 can be established on the server.

15

16 Exemplary Look-Up Process

17 In accordance with one embodiment, a process handles logical ID lookup
18 failures by transferring control to increasingly more exhaustive search processes.
19 The description provided just below expands upon the box 404 in Fig. 4.

20 Fig. 5 illustrates two tables that are generated and maintained in accordance
21 with one embodiment. Table 500 is a canonical table that holds trusted physical
22 ID to logical ID mappings. This table can be maintained on the server side in, for
23 example, database 412 (Fig. 4). Database 412 can also include a second table 502
24 that holds user-provided or user-entered physical ID to logical ID mappings. The
25 user-provided mappings can be those that are entered on the same day. This is

1 advantageous from the standpoint of providing a system that immediately
2 incorporates user feedback for use by others. It will be appreciated that user-
3 provided data is typically less trusted than the trusted data or mappings. To deal
4 with this issue, security measures can be put in place to ensure the integrity of the
5 user-provided data. Exemplary measures are described in detail below in the
6 section entitled "User Feedback Gathering and Management System".

7 When a media's physical ID is received by the server system, a first search
8 is conducted on the trusted table 500. This search looks for a corresponding
9 physical ID that has been mapped to a logical ID. The first search is a low cost
10 search that is configured to search the database quickly. A low cost search can
11 include searches that use a few elements to determine a match. If a matching
12 physical ID to logical ID mapping is found, then the logical ID is used as the basis
13 for a database search to retrieve any relevant metadata. If, on the other hand, a
14 matching physical ID is not found, then a second search is conducted. This second
15 search is conducted on the less trusted table 502 -- the user-provided mappings. If
16 a matching physical ID to logical ID mapping is found, then the logical ID is used
17 as the basis for a database search to retrieve any relevant metadata. The search is
18 desirably another low cost search. If no match is found on this second search, then
19 a third search is conducted back on trusted table 500. This search is a higher cost
20 search that more extensively searches the table. A higher cost search is a search
21 that can use more elements than the low cost search to determine a match. If a
22 match is found, then the corresponding logical ID is used as the basis for a
23 database query. If no match is found, then in one embodiment, the process fails
24 and returns no metadata. The process can also launch into a mode in which the
25 user is prompted to enter information associated with their specific media so that a

1 logical ID can be established for that particular piece of media, and a physical ID
2 to logical ID mapping can be formed. This is part of the processing that takes
3 place using the Wizard, which is discussed in more detail below in the section
4 entitled “Wizard”.

5 The above-described process contributes strongly to the functionality of
6 this embodiment as it not only searches a more extensive set of information for
7 matches, but also allows user-entered data to have an immediate effect on the
8 overall system. If the system fails to find an appropriate match in the user-entered
9 data, it tries a final, more exhaustive search on the canonical table. This fail-over
10 behavior provides very strong performance in the most common case, and varying
11 degrees of slightly degraded performance in the progressively less common cases.
12 The overall result is more immediate physical to logical ID mappings, and more
13 frequent return of metadata for a piece of media.

14 Fig. 6 is a flow diagram that describes steps in a method in accordance with
15 the above-described embodiment. The method can be implemented in any suitable
16 hardware, software, firmware or combination thereof. In the illustrated and
17 described embodiment, the method is implemented in software executing on the
18 server side.

19 Step 600 provides a canonical table with physical ID to logical ID
20 mappings. The mappings contained in this table are trusted mappings. Step 602
21 provides a table with user-provided physical ID to logical ID mappings. This table
22 is not a trusted table. Step 604 receives a physical ID associated with a media
23 such as a CD or DVD. Step 606 conducts a first low cost search of the canonical
24 table, and step 608 determines whether there is a matching physical ID with a
25 logical ID mapping in the canonical table. If there is a matching physical ID, then

1 step 610 uses the logical ID as the basis of a database search and retrieves, formats
2 and returns associated metadata to the user or client computer. If, on the other
3 hand step 608 fails to find a matching physical ID, step 612 conducts a second low
4 cost search of the table containing the user-provided mappings, i.e. the less trusted
5 or untrusted table. Step 614 determines whether there is a matching physical ID
6 with a logical ID mapping in this table. If there is, then step 616 uses the logical
7 ID as the basis of a database search and retrieves, formats and returns associated
8 metadata to the user or client computer. If, on the other hand, step 614 finds no
9 match, step 618 performs a high cost search of the canonical table. Step 620
10 determines whether there is a matching physical ID with a logical ID mapping in
11 the canonical table. If there is, the method branches to step 616. If there is no
12 match, then step 622 fails and returns no metadata. Further processing after
13 failure can include engaging the user via a Wizard for the purpose of establishing a
14 logical ID for the user's specific media.

15

16 **Wizard**

17 In one embodiment, a Wizard is provided to assist in identifying a user's
18 specific media so that a physical ID to logical ID mapping can be made. The
19 Wizard can also assist in establishing a physical ID to logical ID mapping for the
20 user's specific media in the event that the user's specific media cannot be
21 identified. Specifically, recall in Fig. 4 that failing, at 404, to locate a physical ID
22 to logical ID mapping, a Wizard 408 is presented to assist in the discovery process.
23 The discussion below presents but one embodiment of a Wizard that can be used
24 to assist in finding and/or establishing a physical ID to logical ID mapping.

1 Fig. 7 is a flow diagram that describes steps in a method in accordance with
2 one embodiment. The steps can be implemented in any suitable hardware,
3 software, firmware or combination thereof. In the illustrated and described
4 embodiment, the steps are implemented in software. This software can reside on
5 the server side of the system or on the client side of the system. In this particular
6 example, the software resides on the server side of the system, although steps are
7 implemented on both the client and the server side. To this extent, Fig. 7 is
8 divided into two different sections—one labeled “Client” to depict steps that occur
9 on the client side, and one labeled “Server” to depict steps that occur on the server
10 side.

11 Step 700 displays a Wizard UI on a client computer. The Wizard can be
12 implemented in any suitable software. In the illustrated example, the Wizard is
13 implemented using Active Server Pages and DHTML. This step can be
14 implemented, in the described embodiment, upon a failure to find a logical ID
15 associated with a physical ID that has been provided by the client. For example, a
16 user might insert a CD into their computer, where the CD has a physical ID that
17 has not yet been logged in the system. Because the physical ID will not be found
18 by the server, the services of the Wizard will be employed to attempt to identify
19 the specific media so that a logical ID can be associated with its physical ID. This
20 will become more clear as the example of Figs. 8-12 is considered below.

21 Step 702 collects user feedback or information via the Wizard UI. In this
22 step, the Wizard attempts to collect specific information associated with the user’s
23 media. This specific information can include any information that can be used by
24 the system to assist it in establishing a physical ID to logical ID mapping. For
25 example, if the media comprises a music CD, the user can provide feedback in the

1 form of the artist's name or album title. Step 704 then sends the user feedback to
2 the server. Step 706 receives the user feedback and step 708 searches for specific
3 media based on the user's feedback. It will be appreciated and understood that
4 steps 702, 704, 706 and 708 can take place multiple times between the client and
5 server. Such can be the case where, for example, the Wizard uses the user's
6 feedback to provide a number of possible selections, and then progressively
7 narrows down the choices based on the user's additional feedback. This will
8 become evident in the example that follows.

9 If, at step 710, a specific media is found to coincide with the user-provided
10 feedback, then step 712 forms an association between the physical ID associated
11 with the user's media, and the logical ID associated with that media. Now,
12 whenever the user plays that particular piece of media in their media player, the
13 server will be able to use the physical ID to retrieve the associated logical ID, so
14 that it can then query the database(s) that contains the metadata associated with
15 that particular piece of media.

16 If, on the other hand, step 710 is unsuccessful, then step 714 can prompt the
17 user to enter media specific information about their media. Step 716 collects
18 media-specific information from the user. The step can be implemented by the
19 user entering information that, for a CD, can include album title, artist, track titles,
20 and the like. The user can also enter a logical ID that is to be associated with that
21 particular piece of media. Step 718 then sends the media-specific information to
22 the server, and step 720 establishes an association between that particular piece of
23 media and a logical ID for the media, as well as the physical ID for the particular
24 piece of media and the logical ID. It should be noted that steps 714, 716, 718, and
25 720 can be performed when the user is the first user to play their media in an

1 enabled player and the media has not yet been incorporated into the system. For
2 example, if U2's new CD has just been released and has not yet been incorporated
3 into the system (i.e. a logical ID has not yet been established for it), then the first
4 user to play the U2 CD in their enabled player can step through steps 716-718 so
5 that a logical ID can be established on the server for the new CD.

6 Figs. 8-12 show an exemplary Wizard UI 800 that can be provided in one
7 implementation. Fig. 8 shows a start UI that is displayed on a client machine and
8 coincides with step 700 in Fig. 7. Fig. 9 shows a UI in which the user can provide
9 feedback pertaining to an artist's name for a particular CD that they have inserted
10 into their enabled CD player. Once the artist's name is entered, this information is
11 provided to the server which begins searching for a specific media associated with
12 that artist. This coincides with step 708 in Fig. 7. Searching for the specific
13 media can take many forms. Additionally, the searching can involve multiple
14 iterations with the user to narrow down search results. For example, Fig. 10
15 shows a UI that is presented to the user that lists multiple artists that closely match
16 the artist's name entered by the user. Once the user locates their particular artist,
17 they can select the artist. The user's selection is then provided to the server (as by
18 step 704 in Fig. 7) so that the server can narrow down its search somewhat more.
19 Specifically, Fig. 11 shows a UI that is presented to the user. The UI lists various
20 disc titles for the artist that has been confirmed by the user. The user can then
21 select the disc title that corresponds to the CD that they are playing. Once the user
22 selects the appropriate disc title, an association can be made on the server that
23 associates the specific physical ID associated with the user's media, with a
24 corresponding logical ID that can be used as the basis of database searches. Fig.
25

12 is a UI that confirms the track information for the disc title that was selected by
the user.

3

4 **Retrieving, Formatting and Delivering Metadata**

5 After a proper logical ID is found (for example, steps 608, 614, and 620 in
6 Fig. 6), querying code looks for and gathers metadata from the appropriate
7 metadata database(s) based on the logical ID (corresponding to steps 610 and 616
8 in Fig. 6). Any suitable querying code can be used. In the illustrated and
9 described embodiment, the querying code is SQL querying code.

10 The metadata is gathered by procedure code that fetches it from publishing
11 schema for the data. The schema is flattened to some degree and indexed to
12 provide very fast data retrieval.

13 Once the data is gathered, it is formatted for transmission to a client. In the
14 illustrated and described embodiment, formatting takes place via SQLXML (a
15 FOR XML EXPLICIT clause) into one of several extensible metadata delivery
16 schema. These schema vary somewhat based on the type of media that they
17 represent.

18 In the illustrated and described embodiment, there are three XML schema
19 that are provided. Two of the schema are associated with returning metadata
20 associated with CDs, and one of the schema is associated with returning metadata
21 associated with DVDs.

22 XML or “Extensible Markup Language” is a meta-markup language that
23 provides a format for describing structured data. XML is similar to HTML in that
24 it is a tag-based language. By virtue of its tag-based nature, XML defines a strict
25 tree structure or hierarchy. XML is a derivative of Standard Generalized Markup

1 Language (SGML) that provides a uniform method for describing and exchanging
2 structured data in an open, text-based format. XML utilizes the concepts of
3 elements and namespaces. Compared to HTML, which is a display-oriented
4 markup language, XML is a general-purpose language used to represent structured
5 data without including information that describes how to format the data for
6 display.

7 XML “elements” are structural constructs that consist of a start tag, an end
8 or close tag, and the information or content that is contained between the tags. A
9 “start tag” is formatted as “<tagname>” and an “end tag” is formatted as
10 “</tagname>”. In an XML document, start and end tags can be nested within
11 other start and end tags. All elements that occur within a particular element must
12 have their start and end tags occur before the end tag of that particular element.
13 This defines a strict tree-like structure. Each element forms a node in this tree,
14 and potentially has “child” or “branch” nodes. The child nodes represent any
15 XML elements that occur between the start and end tags of the “parent” node.

16 XML accommodates an infinite number of database schemas. Within each
17 schema, a “dictionary” of element names is defined. The dictionary of element
18 names defined by a schema is referred to as a “namespace.” Within an XML
19 document, element names are qualified by namespace identifiers. When qualified
20 by a namespace identifier, a tag name appears in the form
21 “[namespace]:[tagname]”. This model enables the same element name to appear
22 in multiple schemas, or namespaces, and for instances of these duplicate element
23 names to appear in the same XML document without colliding.

24 Start tags can declare an arbitrary number of “attributes” which declare
25 “property values” associated with the element being declared.. Attributes are

1 declared within the start tag using the form “<[tagname]
2 [attribute1],[attribute2]...,[attributeN]>”, where an attribute1 through attributeN
3 are declarations of an arbitrary number of tag attributes. Each attribute declaration
4 is of the form “[attributeName]=[attributeValue]” where each attribute is
5 identified by a unique name followed by an “=” character, followed by the value
6 of the attribute.

7 Within an XML document, namespace declarations occur as attributes of
8 start tags. Namespace declarations are of the form “xmlns:[prefix]=[uri]”. A
9 namespace declaration indicates that the XML document contains element names
10 that are defined within a specified namespace or schema. Prefix is an arbitrary
11 designation that will be used later in the XML document as an indication that an
12 element name is a member of the namespace declared by uri. The prefix is valid
13 only within the context of the specific XML document. “Uri” or universal
14 resource indicator is either a path to a document describing a specific namespace
15 or schema or a globally unique identifier of a specific namespace or schema. Uri
16 is valid across all XML documents. Namespace declarations are “inherited”,
17 which means that a namespace declaration applies to the element in which it was
18 declared as well as to all elements contained within that element.

19 Namespace inheritance within an XML document allows non-qualified
20 names to use “default” namespaces. Default namespaces are explicitly declared as
21 attributes of start tags. Default namespace declarations are of the form
22 “xmlns=[uri]”. Note that the declaration of a default namespace is equivalent to
23 the declaration of a non-default namespace but the prefix is omitted. A namespace
24 specification within an XML document is said to have a “scope” which includes
25 all child nodes beneath the namespace specification.

1

2 First CD XML Schema

3

```
4 <OMI xmlns:sql="urn:schemas-microsoft-com:xml-sql">
5     <name> Back Catalogue</name>
6     <author>Front 242</author>
7     <releasedate>1992 06 02</releasedate>
8     <genre>ROCK </genre>
9     <style>Industrial</style>
10    <rating>6</rating>
11    <copyright></copyright>
12    <label>Wax Trax!</label>
13    <parent>AMG</parent>
14    <track>
15        <name>U-Men</name>
16        <author>Front 242</author>
17    </track>
18 </OMI>
```

19

20

21

22

23

24

25

This particular schema has tags that are associated with the following pieces of information: CD name, author, release date, genre, style, rating, copyright, label, parent, and track (which includes tags for the track name and track author).

16

17 Second CD XML Schema

```
18 <?xml version="1.0" encoding="UTF-8" ?>
19     <OMI xmlns:sql="urn:schemas-microsoft-com:xml-sql">
20         <version>2.0</version>
21             <!-- Description: Version number of this CD OMI
22                 document -->
23         <name>BBC Sessions</name>
24             <!-- Description: Name of album -->
25         <author>Led Zeppelin</author>
26             <!-- Description: Primary Artist or "Various" --
27                 -->
28         <releaseDate>9/18/2000</releaseDate>
29             <!-- Description: Date of initial release -->
30         <label>Warner Brothers</label>
31             <!-- Description: Record label of the release
32                 of this disc -->
33         <genre>ROCK</genre>
```

```

1           <!-- Description: AMG-assigned Genre for album
2           -->
3           <totalLength>74:00</totalLength>
4           <!-- Description: Total program time on disc in
5           mm:ss format -->
6           <style>Rock</style>
7           <!-- Description: AMG-assigned Style -->
8           <rating />
9           <!-- Description: Quality rating: 0= worst,
10          9=best -->
11          <buyURL>http://windowsmedia.com/mediaguide/buynow/buynow.as
12          p? </buyURL>
13          <!-- Description: URL to "buy album" page on
14          windowsmedia.com -->
15          <smallCoverURL>http://services.windowsmedia.com/...
16          </smallCoverURL>
17          <!-- Description: URL to small image (50x50) of
18          cover art -->
19          <largeCoverURL>http://services.windowsmedia.com/...
20          </largeCoverURL>
21          <!-- Description: URL to large image (200x200)
22          of cover art -->
23          <moreInfoURL />
24          <!-- Description: URL to album details page on
25          windowsmedia.com -->
26          <parent>XXX</parent>
27          <!-- Description: Source of data -->
28          <copyright>1995</copyright>
29          <!-- Description: Year of copyright for
30          collection -->
31          <track>
32          <name>You Shook Me</name>
33          <!-- Description: Name of current track -->
34          <author>Led Zeppelin</author>
35          <!-- Description: Primary performer of current
36          track -->
37          <composer>Led Zeppelin</composer>
38          <!-- Description: Primary composer of
39          current track -->
40          </track>
41      </OMI>
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
645
646
647
647
648
649
649
650
651
652
653
654
655
656
656
657
658
658
659
659
660
661
662
663
664
665
666
666
667
668
668
669
669
670
671
672
673
674
675
676
676
677
678
678
679
679
680
681
682
683
684
685
686
686
687
688
688
689
689
690
691
692
693
694
695
696
696
697
698
698
699
699
700
701
702
703
704
705
706
706
707
708
708
709
709
710
711
712
713
714
715
715
716
717
717
718
718
719
719
720
721
722
723
724
725
725
726
727
727
728
728
729
729
730
731
732
733
734
735
735
736
737
737
738
738
739
739
740
741
742
743
744
745
745
746
747
747
748
748
749
749
750
751
752
753
754
755
755
756
757
757
758
758
759
759
760
761
762
763
764
765
765
766
767
767
768
768
769
769
770
771
772
773
774
775
775
776
777
777
778
778
779
779
780
781
782
783
784
784
785
785
786
786
787
787
788
788
789
789
790
791
792
793
794
794
795
795
796
796
797
797
798
798
799
799
800
801
802
803
804
804
805
805
806
806
807
807
808
808
809
809
810
811
812
813
814
814
815
815
816
816
817
817
818
818
819
819
820
821
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
831
832
833
834
834
835
835
836
836
837
837
838
838
839
839
840
841
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
851
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
861
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
871
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
881
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
891
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
911
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
921
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
931
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
941
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
951
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
961
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
991
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1011
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1021
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1031
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1041
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1051
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1061
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1071
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1081
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1091
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1111
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1121
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1131
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1141
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1151
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1161
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1171
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1181
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1211
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1221
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1231
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1241
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1251
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1261
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1271
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1281
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1291
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1311
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1321
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1331
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1341
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1351
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1361
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1371
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1381
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1391
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1411
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1421
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1431
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1441
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1451
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1461
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1471
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1481
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1491
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1511
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1521
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1531
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1541
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1551
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1561
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1571
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1581
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1591
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1611
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1621
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1631
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1641
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1651
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1661
1662
1663
1663
1664
1664
1665
1665
1666
1666
1667
1667
1668
1668
1669
1669
1670
1671
1672
1673
1673
1674
1674
1675
1675
1676
1676
1677
1677
1678
1678
1679
1679
1680
1681
1682
1683
1683
1684
1684
1685
1685
1686
1686
1687
1687
1688
1688
1689
1689
1690
1691
1692
1693
1693
1694
1694
1695
1695
1696
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1703
1704
1704
1705
1705
1706
1706
1707
1707
1708
1708
1709
1709
1710
1711
1712
1713
1713
1714
1714
1715
1715
1716
1716
1717
1717
1718
1718
1719
1719
1720
1721
1722
1723
1723
1724
1724
1725
1
```

1 A “largeCoverURL” tag can be used to provide a URL to a large image of the
2 cover art for a particular CD. Further, a “moreInfoURL” can be used to provide
3 information associated with album details.

4

5 DVD Exemplary Schema

6

```
<?xml version="1.0" encoding="UTF-8" ?>
<DVDData>
<version>1.0</version>
<relation>
    <coverLarge>http://windowsmedia.com/covers/dvd/t00032xurnu.
jpg</coverLarge>
    <coverSmall>http://windowsmedia.com/covers/dvd/t00032xurnu.
jpg</coverSmall>
</relation>
<title>
    <titleNum>1</titleNum>
    <name>Casablanca</name>
    <studio>Warner Home Video</studio>
    <leadPerformer>Humphrey Bogart</leadPerformer>
    <director>Michael Curtiz</director>
    <MPAARating>NR</MPAARating>
    <genre>War Romance</genre>
    <chapter>
        <chapterNum>1</chapterNum>
        <name>Main Title</name>
    </chapter>
</title>
<title>
    <titleNum>2</titleNum>
    <name>Special Features</name>
    <studio>Warner Home Video</studio>
    <leadPerformer>Humphrey Bogart</leadPerformer>
    <director>Michael Curtiz</director>
    <MPAARating>NR</MPAARating>
    <genre>War Romance</genre>
    <chapter>
        <chapterNum>1</chapterNum>
        <name>Casablanca Theatrical Trailer</name>
    </chapter>
</title>
</DVDData>
```

23 In the above DVD schema example, XML tags are provided for various
24 pieces of information that include relation, title, title number, studio, lead
25

1 performer, director, MPAA rating, genre, chapter (including tags for chapter
2 number and chapter name). The relation tag serves as a container for various
3 elements that are “related” to the main element.

4 After the data is formatted into the proper schema, it is sent back over the
5 network to the client application which is then responsible for parsing and
6 rendering the data for the consumer or user.

7 Fig. 13 is a flow diagram that describes steps in a method in accordance
8 with one embodiment. The steps can be implemented in any suitable hardware,
9 software, firmware or combination thereof. In the illustrated and described
10 embodiment, the steps are implemented in software. The steps about to be
11 described are implemented by a server that has access to and searches for metadata
12 associated with a user’s specific media.

13 Step 1300 searches a metadata database using a logical ID that is associated
14 with a user’s media. This step is implemented after a suitable physical ID to
15 logical ID mapping has been found. Any suitable search can be conducted. In the
16 illustrated and described embodiment, the search is a SQL search. Step 1302
17 retrieves metadata associated with the user’s media. The metadata that is retrieved
18 can be any suitable metadata that is provided for the media. Examples of metadata
19 are given above. Step 1304 formats the metadata using an XML schema. Any
20 suitable XML schema can be used. Three exemplary non-limiting XML schemas
21 are given above. XML schemas are characterized by associated tags. The XML
22 tags can be associated with any type of metadata that is or can be envisioned to be
23 provided in connection with a particular type of media. Once the metadata is
24 formatted in an appropriate XML schema, step 1306 sends the formatted metadata
25 to a client.

1 Fig. 14 is a flow diagram that describes steps in a method in accordance
2 with one embodiment. The steps can be implemented in any suitable hardware,
3 software, firmware or combination thereof. In the illustrated and described
4 embodiment, the steps are implemented in software. The steps about to be
5 described are implemented by a client that receives the formatted data in
6 accordance with Fig. 13.

7 Step 1400 receives XML-formatted metadata. Step 1402 parses the XML-
8 formatted metadata. Parsing techniques are well-known in the art and,
9 accordingly, are not described in additional detail here. Once the XML data is
10 parsed, step 1404 renders and displays the metadata on the client for the user.

11 It should be appreciated that the use of XML in the process of formatting
12 and returning metadata to the user is advantageous from a number of perspectives.
13 For example, XML provides a mechanism by which a robust collection of
14 metadata can be returned to the user. Thus, the user experience can be greatly
15 enhanced. Additionally, by virtue of its extensibility, XML provides a mechanism
16 that allows metadata databases to be flexibly managed to accommodate new and
17 different types of metadata. This is highly desirably given the likelihood that
18 metadata types will continue to evolve into the future.

19

20 **User Feedback Gathering and Management System**

21 In accordance with one embodiment, user-submitted information can be
22 integrated into various data stores in a secure and automated way. Recall from the
23 above discussion that users can, in some instances, provide their own data that can
24 be used to establish physical ID to logical ID mappings. This embodiment
25 recognizes that no collection of physical ID to logical ID mappings will ever be

1 absolutely complete. Accordingly, a mechanism is provided by which a single
2 user can contribute a mapping fairly effortlessly to the system.

3 It should be appreciated that any system which empowers users in this
4 manner necessarily opens itself up to significant risks of tampering, erroneous
5 entries, and other security concerns. In the discussion below, an approach is
6 provided that deals with automatically analyzing user-provided data to be
7 incorporated in such a way that it is reasonably certain to be accurate, and is self-
8 correcting in the case where it is not accurate.

9 As an overview, consider the following. When the server system initially
10 fails to retrieve an appropriate logical ID, the system can further prompt the user
11 to attempt to make a logical-physical match by searching the database of all
12 possible choices. Examples of this have been given above. If a search is
13 successful, and the user confirms a match, that information is entered into a
14 holding area which will be referred to as a “user feedback table”. Information in
15 the user feedback table is immediately incorporated into all future physical-logical
16 ID mapping attempts. This means that a new mapping is available to all other
17 users of the system immediately upon entry. This results in an immediate
18 potential increase in metadata returns.

19 Because of the method of operation of the fail-over query system, the user-
20 entered mapping is immediately available for consumption if no previous mapping
21 for that physical ID exists in the canonical table. This corresponds to box 414 in
22 Fig. 4.

23 In addition to the immediate feedback operation, the contents of the user
24 feedback table are processed or evaluated by a statistical system, on a regular
25 basis, to help ensure the overall quality and reliability of the mapping data. This

1 statistical system processes the user feedback table on a periodic basis (i.e. every
2 night), and processes each and every physical ID that has been entered since its
3 last runtime. For each physical ID, it queries the entirety of previously entered
4 user feedback for that physical ID, as well as the current canonical mapping of that
5 physical ID. It weights the canonical mapping with an appropriate variable weight
6 (e.g. 50 times a normal mapping), and adds it to a statistical distribution of
7 physical-logical ID mappings from the user feedback table. If any one single
8 mapping exceeds a definable threshold (e.g. receives 50% or more of the total
9 weighted distribution), then the mapping that exceeds the threshold is published
10 into the canonical table. Conversely, if no single mapping exceeds the definable
11 threshold (e.g. receives 50% or more of the total weighted distribution), then the
12 mapping is in question and is removed from the lookup table.

13 Consider, for example, Fig. 15. There, database 412 contains a user
14 feedback table 1500. This table can be very large (e.g. millions and millions of
15 rows long) and contains potentially valid physical ID to logical ID mappings that
16 were entered by users. The server system can process table 1500 on a nightly
17 basis so that it can be cycled into the trusted canonical table discussed above.

18 In the illustrated and described user feedback table, column 1502 is
19 associated with physical IDs, column 1504 is associated with logical IDs to which
20 corresponding physical IDs are mapped, and column 1506 is associated with a
21 timestamp. The timestamp for each mapping indicates the time when it was
22 entered into the user feedback table 1500. On a regular basis, the system
23 processes the physical IDs that have been entered by users for purposes of
24 ascertaining whether they should be entered into the canonical table. This process
25 also desirably serves to filter erroneous or spuriously-entered mappings.

1 Fig. 16 is a flow diagram that describes steps in a method managing user-
2 provided entries in accordance with one embodiment. The steps can be
3 implemented in any suitable hardware, software, firmware or combination thereof.
4 In the illustrated and described embodiment, the steps are implemented in
5 software.

6 Step 1600 computes a list of physical IDs that have been entered since the
7 system last ran a screening process on the user entries and which are to be
8 statistically evaluated. This removes any duplicates that might appear in table
9 1500. The result of this computation is another table, such as the one shown at
10 1700 in Fig. 17. Once table 1700 is provided, step 1602 computes, for each
11 physical ID appearing in table 1700, the total number of user feedback entries to
12 date and the logical IDs that have been provided for each physical ID. Consider
13 again Fig. 17 where table 1702 is shown. This table contains all of the user
14 feedback entries for physical ID “1”, as well as the corresponding logical IDs to
15 which the physical IDs have been mapped. Notice that the physical ID of 1 has
16 been mapped to various logical IDs that include 20, 3, and 15. These different
17 mappings can be the result of user error, or can be an attempt by malicious
18 individuals to provide erroneous mappings.

19 Next, step 1604 computes a distribution of user-entered logical IDs for the
20 particular physical ID of interest. Consider again Fig. 17 and table 1704. There, a
21 column designated “Logical ID” contains each logical ID for the particular
22 physical ID of interest, and a column designated “Dist. Count” contains, for each
23 logical ID, a value associated with the number of times its corresponding logical
24 ID has been mapped to by a physical ID. So, for example, from table 1704, it
25 appears that LID 20 was mapped to twice, as was LID 3. LID 15 was mapped to

1 once. Table 1704 also contains a column designated “%”, the purpose of which
2 will become evident below.

3 Once the distribution has been calculated by step 1604, step 1606 adds a
4 weighted entry to table 1704 based on the current canonical match for the logical
5 ID that corresponds to the physical ID of interest. Specifically, in this example,
6 assume that the physical ID undergoing processing (i.e. PID = 1) has a mapping in
7 the canonical table that corresponds to a logical ID of 13. Thus, an entry is made
8 in table 1704 to contain the LID 13, as well as a weighted value that is selected in
9 accordance with some definable criteria. In this example, assume that a weighted
10 value of “50” is adequate.

11 From there, steps 1608 and 1610 compute the most likely physical ID to
12 logical ID match using the data in table 1704. Any suitable algorithm can be used
13 for computing the most likely match. In but one example, a standard majority
14 algorithm is used. Specifically, in this example, the distribution count is totaled
15 for a sum of 55. Now, step 1608 computes a percentage for each table entry
16 (corresponding to the “%” column). Specifically, for LID 20, the percentage is
17 2/55; for LID 3 the percentage is 2/55; for LID 15 the percentage is 1/55; and for
18 LID 13, the percentage is 50/55. Once the percentage is computed, step 1610
19 selects the LID for which the corresponding percentage meets predefined criteria.
20 For example, for a particular logical ID to be selected, the predefined criteria
21 might be that the percentage has to be greater than 0.50. Once the logical ID has
22 been selected for a corresponding physical ID, step 1612 determines whether the
23 selected logical ID is different from the logical ID that appears in the canonical
24 table. If the logical ID is different, then step 1614 updates the canonical table with
25 the logical ID that most likely matches the corresponding physical ID. If, on the

1 other hand, the selected logical ID is the same as the logical ID that appears in the
2 canonical table, step 1616 maintains the canonical table entry for the physical ID.

3 Step 1618 determines whether there are any additional physical IDs in table
4 1700 that need to be processed. If there are, the method branches back to step
5 1602. If there are not, the method ends at step 1620 until it is time to again
6 process the user-provided physical ID to logical ID mappings.

7

8 Conclusion

9 The systems and methods described above can greatly enhance the user's
10 media experience when they play a particular piece of media in an enabled player.
11 A robust collection of metadata is available for provision to the user through the
12 use of an organized mapping process that maps physical IDs associated with the
13 user's media to logical IDs that are used for database queries. A data return
14 mechanism and schema set provide a rich and extensible mechanism that
15 facilitates data exchange between a client and a server. A user feedback system
16 ensures that users can contribute to a set of understood physical ID to logical ID
17 mappings, and that the contributions are immediately available for use by others.
18 Management processes ensure that user contributions are understood, and that
19 such contributions can be trusted.

20 Although the invention has been described in language specific to structural
21 features and/or methodological steps, it is to be understood that the invention
22 defined in the appended claims is not necessarily limited to the specific features or
23 steps described. Rather, the specific features and steps are disclosed as preferred
24 forms of implementing the claimed invention.